

IN THE CLAIMS

1. (Original): A server system utilizing an HttpSession object in a Java servlet application program interface (API) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory;

a second memory having a database for storing HttpSession objects for http sessions being handled by said JVMs, said memory being accessible by each of said JVMs;

a first computer program adapted to store in a memory local to said server running said JVM HttpSession data for each http session handled by said JVM;

a second computer program adapted to write to said database a copy of said HttpSession data for each said http session at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session.

2. (Original): The server system of claim 1 further comprising a third computer program adapted to write to said database a copy of said HttpSession object data for each said http session at the time the http session is initiated.

3. (Original): The server system of claim 2 wherein said plurality of JVMs are running on a plurality of servers.

4. (Original): The server system of claim 3 wherein said writes to said database are performed at the end of a corresponding servlet service method.

5. (Original): The server system of claim 4 wherein said server system services the World Wide Web.

6. (Original): The server system of claim 5 wherein said Java servlet APIs are J2EE servlet APIs.

7. (Original): The server system of claim 1 wherein said second program polls said session objects stored in said memories local to said JVMs to determine if said predetermined time interval has passed since they have been updated and wherein said second program is adapted to write to said database only copies of said HttpSession objects that have been updated within said predetermined time interval.

8. (Original): The server system of claim 7 wherein said second computer program is invoked at predetermined intervals.

9. (Original): The server system of claim 1 wherein said time interval is configurable.

10. (Original): The server system of claim 9 wherein said time interval is between ten seconds and five minutes.

11. (Original): A method of maintaining session data in a server system servicing a network, said server system maintaining state data pertaining to sessions, said method comprising the steps of:

(1) storing data for each session in a memory local to a server servicing said session;

(2) writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated

times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions.

12. (Original): The method of claim 11 further comprising the step of:

(3) writing in said database a copy of said session data for each said http session at the time the http session is initiated.

13. (Original): The method of claim 11 wherein said server system services the World Wide Web.

14. (Original): The method of claim 11 wherein said server system comprises a plurality of Java Virtual Machines (JVMs) running on a plurality of servers, and wherein said data for said sessions comprises an HttpSession object of a Java servlet application program interface (API).

15. (Original): The method of claim 14 wherein said Java servlet APIs are J2EE servlet APIs.

16. (Original): The method of claim 11 wherein said time interval is configurable.

17. (Currently Amended): The method of claim ~~14~~14 further comprising the step of:

(4) polling said session objects stored in said memories local to said JVMs to determine if they have been updated since the last time step (2) was performed; and wherein, in step (2), only copies of said HttpSession objects that have been updated within said predetermined time interval are written to said database.

18. (Original): A server system utilizing HttpSession objects in a Java servlet application program interface (API) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory;

a memory having a database for storing HttpSession objects for http sessions being handled by said plurality of JVMs, said memory being accessible by each of said JVMs;

a first computer program adapted to store in a memory local to said server running said JVM HttpSession object data for each http session handled by a JVM;

a second computer program adapted to write a copy of said HttpSession data for each said http session in said database at designated times, said designated times determined as a function of at least one of (a) the number of times the HttpSession object data is updated in said local memory and (b) the number of times an http request in said http session is serviced.

19. (Original): The server system of claim 18 wherein said second computer program is adapted to write said HttpSession object data to said database after X HttpSession updates in said local memory, where X is an integer greater than or equal to 2.

20. (Original): The server system of claim 18 wherein said second computer program is adapted to write said HttpSession object data to said database after X http requests in said http sessions, where X is an integer greater than or equal to 2.

21. (Currently Amended): The server system of claim ~~47~~ 18 further comprising a third computer program adapted to store in said database a copy of said HttpSession object data for each said http session at the time the http session is created.

22. (Original): The server system of claim 21 wherein said plurality of JVMs are running on a plurality of servers.

23. (Original): The server system of claim 22 wherein said Java servlet APIs are J2EE servlet APIs.

24. (Original): The server system of claim 18 wherein said writes to said database are performed at the end of a first servlet service method of a corresponding http session received after said designated time.